

[illegible]

FIELD OF THE INVENTION

[0001] This invention relates generally to computer firmware or operating systems and more particularly to shortening computer boot-up time.

BACKGROUND OF THE INVENTION

[0002] During the boot process of many computer systems, the firmware or operating system spends time discovering the amount of memory, configuring memory, walking I/O busses to discover devices (LAN, PCI, SCSI, I2C, etc.), configuring devices, and discovering and configuring the machines topology. Unfortunately, these tasks may take a long time since the firmware or operating system must instruct the computer to probe many addresses and compare many of the results against a relatively large database of device characteristics just to complete the discovery portions of these tasks.

[0003] The length of time these tasks take impacts the overall length of time it takes to boot. Slow boot times may aggravate users and increase the time upgrades requiring a reboot take to complete. Slow boot times may also give a user the impression that the computer system is slow or is of inferior speed.

[0004] Accordingly, there is a need in the art for methods and apparatus that reduce the boot times of computer systems.

SUMMARY OF THE INVENTION

[0005] The invention reduces the time it takes to complete a boot process by reducing or eliminating the need to discover many or all of the devices and machine topology. When the computer is in standby, access panels are monitored so that it is detected when they are opened. If an access panel has not been opened, then it is assumed that the system configuration, devices, etc. that may have been affected by access into that component have not changed since the last boot process and there is no need to spend time discovering devices and machine topology that may have been changed by access to that panel. Stored, instead of discovered, configuration information is then used to configure those devices. This reduces the time required to boot since discovery is not performed on all of the devices in the computer. A service processor or other logic may be used to detect which access panels are opened. Also, those elements of the configuration that may have changed by the particular access panels that were opened are re-discovered and that information may be stored for future use during future computer boot-ups.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1 is a state transition illustration illustrating a state machine that is an example implementation of reduced time boot-ups.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0008] FIG. 1 is a state transition illustration illustrating a state machine that is an example implementation of reduced time boot-ups. The state of the system is determined by the state of the system power supplies and two sets of variables COMPONENTS and INTRUSIONS. COMPONENTS represents a set of subsystems or devices that require discovery and configuration. Thus, there is a member in the set COMPONENTS for each subsystem or device that has been discovered or is known to inherently exist in the system. Each member of the set COMPONENTS may have three values: 1) NO_INTRU_SUPPORT for a component that does not support intrusion detection or must be re-detected and re-configured during every boot regardless of any changes to the system; 2) INTRU_AND_NOTVALID for a component that does support intrusion detection but does not have a valid profile associated with it, or, 3) INTRU_AND_VALID for a component that supports intrusion detection and has a valid profile associated with it. Hereinafter, an individual and arbitrary member of the set COMPONENTS may be referred to as COMPONENT_x.

[0009] INTRUSIONS represents a set of flags that signal intrusion detection for the elements in the COMPONENTS set that support intrusion detection. Each member of the set INTRUSIONS can be either asserted or deasserted. Asserted meaning that either an intrusion has occurred on the corresponding component in COMPONENTS, or it is not known if an intrusion has occurred or not occurred. A member of INTRUSIONS being deasserted means that no intrusion has occurred into the corresponding component. Thus each element in the set INTRUSIONS corresponds to an element in COMPONENTS that supports intrusion detection. .

Hereinafter, an individual and arbitrary member of the set INTRUSIONS may be referred to as INTRUSION_y.

[0010] In FIG. 1, State 0 **110** represents total system shutdown with no power to the system. This includes no power to both the main system and any standby power that may preserve the variables storing the members of the set INTRUSIONS and the variable storing the members of the set COMPONENTS variables, and any logic or service processor that can detect intrusions.

[0011] State 1 **111** represents the situation when the standby power is on and the primary system power is off. When standby power is on it allows a service processor or other logic to monitor intrusions into access panels. Accordingly, State 1 **111** is reached from State 0 **110** by transition **1101** when system standby power is turned on.

[0012] In FIG. 1, when standby power is turned on a transition from State 0 **110** to State 1 **111** via transition **1101** is made and all the members of INTRUSIONS are asserted. This will force a complete scan of all components to take place during the boot process since nothing will be assumed about the physical characteristics or state of the computer.

[0013] When primary system power is turned on a transition from State 1 **111** transitions to State 2 **112** via transition **1112** is made and a first member of COMPONENTS, COMPONENT_x, corresponding to the Base Component is created. The value of COMPONENT_x is set to either INTRU_AND_NOTVALID or NO_INTRU_SUPPORT depending upon whether the Base Component supports intrusion detection or not, respectively. The Base Component is simply the first element listed in the set of components COMPONENTS and may be the main chassis or other major component that houses a group of devices or other components.

transition to State 2 **112** (via transition **1152**) is then made and another member of COMPONENTS is made the current component for consideration in State 2 **112**.

[0017] If a transition to State 4 **114** (via transition **1124**) is made, then discovery and configuration code is executed for the Base Component and a valid profile for the Base Component is built. This also may result in additional components being discovered. When additional components are discovered, additional elements are added to the set COMPONENTS and their values set to NO_INTRU_SUPPORT, INTRU_AND_NOTVALID, and INTRU_AND_VALID according to what was discovered or already known about them. A transition to State 2 **112** (via transition **1142**) is then made and another member of COMPONENTS is made the current component for consideration in State 2 **112**.

[0018] Once back in State 2 **112**, after returning from State 4 **114** or State 5 **115**, a component other than the Base Component is now the current component (provided new components have been discovered in either State 4 **114** or State 5 **115**). The next transition out of State 2 **112** is to either State 5 **115** via transition **1125** or to State 4 **114** via transition **1124** because all the members of INTRUSIONS were asserted in during transition **1102** and have not yet been deasserted. In State 4 **114** or State 5 **115**, discovery, configuration, and possibly profile building for the current component takes place. Therefore, upon returning from either of these states, new components may have been added to COMPONENTS. Also, another member of COMPONENTS that has not been the current member (if any) will have been selected as the current component. This process repeats itself for each member in or added to COMPONENTS until there are no more components in need of discovery and configuration. When there are no more components in need of discovery and

configuration, a transition to State 6 116 via transition 1126 is made. In State 6 116, the boot process is completed and then followed by normal operation.

[0019] A transition out of State 6 116 is made when one or both of the power supplies are switched off. If both primary system power and standby power are turned off, then a transition back to State 0 110 via transition 1106 is made. If only the primary system power is turned off, a transition to State 1 111 via transition 1161 is made and all the members of INTRUSIONS are deasserted. While the system is in State 1 111, access panels may be opened and various components added or removed. When an access panel is opened, this event is noted by the service processor or other logic and the appropriate member of INTRUSIONS is asserted.

[0020] When primary system power is once again turned on a transition from State 1 111 transitions to State 2 112 via transition 1112 is made and the current component is set to the member of COMPONENTS corresponding to the Base Component. From State 2 112, there may be a transition to either State 3 113, State 4 114 or State 5 115 depending upon the value of the current member of COMPONENTS and the value of the member of INTRUSIONS corresponding to that member, or to State 6 116 when there are no more components to consider. The transitions from State 2 112 to State 4 114, State 5 115, and State 6 116 were described, above.

[0021] If a valid profile for the current component exists, and the current component supports intrusion detection (i.e. COMPONENT_x is set to INTRU_AND_VALID) and the corresponding member of INTRUSIONS is deasserted for this component, then a transition from State 2 112 to State 3 113 is made via transition 1123. In State 3 113, the current component is initialized and configured using the stored profile that was created for it in State 4 114. After this is

complete, a transition from State 3 113 back to State 2 112 (via transition 1132) is made and another member of COMPONENTS that has not been the current member (if any) is selected as the current component. Note that no discovery is done in State 3 113. This helps speed the boot process.

[0022] The process of transitioning out of State 2 112 to either State 3 113, State 4 114, or State 5 115 repeats itself for each member in or added to COMPONENTS until there are no more components in need of discovery, initialization, or configuration. When there are no more components in need of discovery, initialization, or configuration, a transition to State 6 116 via transition 1126 is made. In State 6 116, the boot process is completed and then followed by normal operation.

[0023] In a typical system, the code executed in State 3 113 completes in less time than the code executed for the same component in either State 4 114 or State 5 115 because the code executed in State 3 113 is able to configure the component using stored profile information instead of having to execute additional code to discover information about the component. Accordingly, once there are stored profiles for one or more components, and none of the access panels have been opened (which may cause a change in one or more stored profiles) the boot time is reduced. Since many computer boot-ups are for reasons that do not require opening of access panels, these boot-ups may occur significantly faster.

[0024] In addition to the transitions described above, State 1 111 may be reached from State 2 112, State 3 113, State 4 114, or State 5 115 by transitions 1121, 1131, 1141, 1151 which represent the case when primary power is turned off and standby power is turned on during those states. Likewise, but not shown in FIG. 1, State 0 may be reached from any other state when primary power and standby power are both switched off.

[0025] Although a specific embodiment of the invention has been described and illustrated, the invention is not to be limited to the specific forms or arrangement so described and illustrated. The invention is limited only by the claims.

09871329:053001
100651:0221860